

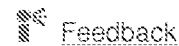

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

(Title:refactoring)



THE ACM DIGITAL LIBRARY



(Title:refactoring)

Term used: refactoring

Found 90 of 240,155

Sort results
by
☒ [Save results to a Binder](#)

 Refine these results with [Advanced Search](#)
Display
results
☐ [Open results in a new window](#)

 Try this search in [The ACM Guide](#)

Results 1 - 20 of 90

 Result page: 1 2 3 4 5 [next](#) [>>](#)

1 [Refactoring-aware version control](#)



Tammo Freese

 May 2006 I CSE '06: Proceedings of the 28th international conference on
Software engineering

Publisher: ACM

 Full text available: [pdf\(180.54 KB\)](#) [Additional Information: full citation, abstract, references, index terms](#)

Today, refactorings are supported in some integrated development environments (IDEs). The refactoring operations can only work correctly if all source code that needs to be changed is available to the IDE. However, this precondition neither holds for ...

Keywords: application programming interface, eclipse, library, refactoring, software evolution, subversion, version control

Ads by

[Optical Switching](#)
24x24 to 190x190
optical switches
Free online demo
on how it works
[Glimmerglass.com/Optical](#)

[Support101](#)
Providing computer
services throughout
the Maryland area.
[www.support101.com](#)

2 [Refactoring for generalization using type constraints](#)



Frank Tip, Adam Kiezun, Dirk Bäumer

 October 2003 OOPSLA '03: Proceedings of the 18th annual ACM SIGPLAN
conference on Object-oriented programing, systems,
languages, and applications

Publisher: ACM

 Full text available: [pdf\(423.08 KB\)](#) [Additional Information: full citation, abstract, references, cited by, index terms](#)

Refactoring is the process of applying behavior-preserving transformations (called "refactorings") in order to improve a program's design. Associated with a refactoring is a set of preconditions that must be satisfied to guarantee that program behavior ...

Keywords: class hierarchy, program analysis, refactoring, subtyping, type constraints

[Storage Area Networks](#)
White Paper. SAN
Solution Overview
Free PDF
Download: Ciena
[www.ciena.com](#)

[The Wyless Network](#)
1 World 1 Network
1 SIM 1 Tariff The
global wireless data
network
[www.wyless.com](#)

3 [An innovative approach to teaching refactoring](#)

Suzanne Smith, Sara Stoecklin, Catharina Serino

March 2006 ACM SIGCSE Bulletin, Volume 38 Issue 1



Publisher: ACM

Full text available: pdf(73.54 KB) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

Refactoring is the process of transforming the internal structure of existing code while keeping the integrity of the code's functional requirements. Refactoring is proven to increase program maintainability, flexibility, and understandability and is ...

Keywords: computer science curriculum, extreme programming, object-oriented design, refactoring, software engineering

4 [Beyond refactoring: a framework for modular maintenance of crosscutting design idioms](#)



Macneil Shonle, William G. Griswold, Sorin Lerner

September 2007 ESEC-FSE '07: Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering

Publisher: ACM

Full text available: pdf(350.51 KB) Additional Information: [full citation](#), [abstract](#),
[references](#), [index terms](#)

Despite the automated refactoring support provided by today's IDEs many program transformations that are easy to conceptualize--such as improving the implementation of a design pattern--are not supported and are hence hard to perform. We propose an extension ...

Keywords: design patterns, refactoring

5 [Automated Inference of Pointcuts in Aspect-Oriented Refactoring](#)

Prasanth Anbalagan, Tao Xie

May 2007 ICSE '07: Proceedings of the 29th International Conference on Software Engineering

Publisher: IEEE Computer Society

Full text available: pdf(316.63 KB) Additional Information: [full citation](#), [abstract](#),
[references](#), [index terms](#)

Software refactoring is the process of reorganizing the internal structure of code while preserving the external behavior. Aspect-Oriented Programming (AOP) provides new modularization of software systems by encapsulating crosscutting concerns. Based ...

6 [Clone detection and refactoring](#)



Robert Tairas

October 2006 OOPSLA '06: Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications

Publisher: ACM

Full text available: pdf(170.58 KB) Additional Information: [full citation](#), [abstract](#),
[references](#), [index terms](#)

The area of clone detection (i.e., searching for duplicate fragments of

source code) has received wide interest recently as indicated by numerous efforts in clone detection tool development. Additionally, some work has been done in the area of selecting ...

Keywords: clone detection, duplicate code, refactoring

7 [An approach to aspect refactoring based on crosscutting concern](#)



[types](#)

Marius Marin, Leon Moonen, Arie van Deursen

May 2005 MACS '05: Proceedings of the 2005 workshop on Modeling and analysis of concerns in software

Publisher: ACM

Full text available: [pdf\(153.93 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

We argue for the importance of organizing generic crosscutting concerns by distinctive properties and describing them as *types*. A *type*'s properties consist of a general intent, an implementation idiom criteria, and one (desired) aspect ...

8 [Feature oriented refactoring of legacy applications](#)



Jia Liu, Don Batory, Christian Lengauer

May 2006 I CSE '06: Proceedings of the 28th international conference on Software engineering

Publisher: ACM

Full text available: [pdf\(302.76 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Feature oriented refactoring (FOR) is the process of decomposing a program into features, where a feature is an increment in program functionality. We develop a theory of FOR that relates code refactoring to algebraic factoring. Our theory explains relationships between ...

Keywords: feature interactions, features, product lines, program algebra, program synthesis, refactoring

9 [Work experience versus refactoring to design patterns: a controlled experiment](#)



T. H. Ng, S. C. Cheung, W. K. Chan, Y. T. Yu

November 2006 SIGSOFT '06/ FSE-14: Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering

Publisher: ACM

Full text available: [pdf\(396.39 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Program refactoring using design patterns is an attractive approach for facilitating anticipated changes. Its benefit depends on at least two factors, namely the effort involved in the refactoring and how effective it is. For example, the benefit would ...


Keywords: controlled experiment, design patterns, refactoring

10 [Refactoring model-view-controller](#)

Stuart Hansen, Timothy V. Fossum

October 2005 Journal of Computing Sciences in Colleges, Volume 21 Issue 1

Publisher: Consortium for Computing Sciences in Colleges

Full text available:  [pdf\(479.32 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Model-view-controller (MVC) is an important architectural design pattern that frequently does not receive the attention it deserves. It is generally treated as a prescriptive design pattern, where students are taught to use three different categories ...


11 [Using refactoring and unification rules to assist framework evolution](#)



Mariela Cortes, Marcus Fontoura, Carlos Lucena

November 2003 ACM SIGSOFT Software Engineering Notes, Volume 28 Issue 6

Publisher: ACM

Full text available:  [pdf\(309.29 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Although object-oriented software development has experienced the benefits of using frameworks, a thorough understanding of how to change them to meet evolving requirement needs is still object of research. Therefore framework development is very expensive, ...

Keywords: framework redesign and evolution, object-oriented frameworks, refactoring, unification rules


12 [Refactoring](#)



Martin Fowler

May 2002 I CSE '02: Proceedings of the 24th International Conference on Software Engineering

Publisher: ACM

Full text available:  [pdf\(88.84 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This tutorial is an example driven introduction to refactoring: a disciplined approach to changing the design of an existing code base.


13 [Towards scalable compositional analysis by refactoring design models](#)



Yung-Pin Cheng, Michal Young, Che-Ling Huang, Chia-Yi Pan

September 2003 ACM SIGSOFT Software Engineering Notes, Volume 28 Issue 5

Publisher: ACM

Full text available:  [pdf\(216.81 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index](#)

[terms](#)

Automated finite-state verification techniques have matured considerably in the past several years, but state-space explosion remains an obstacle to their use. Theoretical lower bounds on complexity imply that all of the techniques that have been developed ...

Keywords: CCS, compositional analysis, promela, refactoring

14 [Refactoring design models for inductive verification](#)



Yung-Pin Cheng

July 2002 I SSTA '02: Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis

Publisher: ACM

Full text available: [pdf\(547.36 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#)

Systems composed of many identical processes can sometimes be verified inductively using a network invariant, but systems whose component processes vary in some systematic way are not amenable to direct application of that method. We describe how variations ...

Keywords: compositional analysis, concurrency, network invariants, parameterized system, refactoring

15 [Tools for a successful refactoring](#)



Emerson Murphy-Hill, Andrew P. Black

October 2006 OOPSLA '06: Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications

Publisher: ACM

Full text available: [pdf\(42.53 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Although refactoring tools have been integrated into numerous development environments over the past ten years, we have seen little variation in the human interfaces to refactoring tools -- when a refactoring fails, most tools present a textual error ...

Keywords: environments, refactoring, tools, usability

16 [Automated method-extraction refactoring by using block-based slicing](#)



Katsuhisa Maruyama

May 2001 SSR '01: Proceedings of the 2001 symposium on Software reusability: putting software reuse in context

Publisher: ACM

Full text available: [pdf\(174.09 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Refactoring improves the design of existing code but is not complex to do by hand. This paper proposes a mechanism that automatically refactors methods of object-oriented programs by using program slicing. To restructure a method without changing its ...

17 [Refactoring support for class library migration](#)

Ittai Balaban, Frank Tip, Robert Fuhrer

October 2005 ACM SIGPLAN Notices, Volume 40 Issue 10

Publisher: ACM

Full text available: [pdf\(309.81 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

As object-oriented class libraries evolve, classes are occasionally deprecated in favor of others with roughly the same functionality. In Java's standard libraries, for example, class Hashtable has been superseded by HashMap, and Iterator is now preferred ...

18 [Refactoring the JUnit framework using aspect-oriented programming](#)

Uirá Kulesza, Cláudio Sant'Anna, Carlos Lucena

October 2005 OOPSLA '05: Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications

Publisher: ACM

Full text available: [pdf\(148.78 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [index terms](#)

Aspect-Oriented Programming (AOP) has been proposed as a complementary modularization technique to the existing object-oriented techniques. AOP encourages the modular development of complex software by providing support for cleanly separating the basic ...

Keywords: aspect-oriented programming, refactoring

19 [A refactoring tool for Ada 95](#)

Paul Anderson

December 2004 ACM SIGAda Ada Letters, Volume XXIV Issue 4

Publisher: ACM

Full text available: [pdf\(226.94 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [index terms](#)

Refactoring is a technique for restructuring code to improve its design. A tool for automatically applying refactoring transformations to Ada 95 programs is described. The tool is based on a language-neutral static-analysis toolset named CodeSurfer, ...

Keywords: ASIS, Ada 95, refactoring, software engineering tools, static analysis

20 [Automated refactoring to introduce design patterns](#)

Mel Ó Cinnéide

June 2000 I CSE '00: Proceedings of the 22nd international conference on Software engineering

Publisher: ACM

Additional Information: [full citation](#), [abstract](#),

Full text available:  pdf(96.87 KB)

[references](#), [cited by](#), [index terms](#)

Software systems have to be flexible in order to cope with evolving requirements. However, since it is impossible to predict with certainty what future requirements will emerge, it is also impossible to know exactly what flexibility to build into a system. ...

Keywords: behaviour preservation, design pattern, program refactorings

Results 1 - 20 of 90

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [next](#) [>>](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2008 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)